Chapter 7.
Using Options

Topic: *Modeler Control

Options are global variables that may be used to modify the behavior of ACIS. Options are documented in reference templates in online help. Refer to the *3D ACIS Online Help User's Guide* for a description of the fields in the option reference template.

Many options are only for internal testing by *Spatial* and are not to be modified by users. If an option is not documented in online help, its value should not be modified by users. Some internal use options are documented in reference templates that contain only a statement indicating that they should not be modified.

In general, options are valid in two contexts: Scheme, and a C++ application. However, some options may not apply to all contexts. The options available to a particular application depend on the component libraries linked into the executable.

Options are defined in the ACIS code using the C++ class option_header. An option has a *name string* and a *value*. An application may inquire about an option's current setting or modify an option's value.

The name string is used to identify the option when setting its value, performing an inquiry on the option, or displaying its value. The name string is the same in both contexts (Scheme, or C++). The name string for some options may be abbreviated when used to set the option.

For many options, the value simply indicates an *on* or *off* state. For other options, the value may be a number (either an integer or a floating point number) or character string.

The data type and representation of an option's value vary depending on the context. For example, an option that just has an on/off state is represented in Scheme with the Scheme data type boolean, with possible values #t for on or #f for off. And in C++, this same option is represented as a logical (which is defined in ACIS and is equivalent to an int) with the values TRUE for on or FALSE for off (the symbolic constants TRUE and FALSE are defined as equivalent to the integer constants 1 and 0, respectively).

An option is set in Scheme using the Scheme extension option:set; or in a C++ application using an API function. There are three APIs available for setting options in C++; the appropriate API to use depends on the data type of the option value:

api_set_int_option . . . . . when the value is an int

api_set_dbl_option . . . . . when the value is a double

api_set_str_option . . . . . when the value is a char* (character string)